KNC-D：地理情報システムのためのバッファ管理手法

6W－3

能登谷淳一，陳漢雄，大保信夫
筑波大学

1　はじめに

計算機の処理能力の向上に伴い，様々な応用分野における計算機の高度な利用が進んでいる．特に近年，環境問題などへの対応の必要性から，地理情報システム (GIS) に対する要求が増大している [2]．このような状況下，地理情報の複雑化に伴い，GIS におけるデータベース技術の積極的な利用が望まれているが，一方で，GIS の性能向上のためには，地理情報特有の性質を効果的に利用する必要がある．空間中の航行は GIS における主要な問い合わせパターンの一つであるが，このような問い合わせでは，地理情報特有のセマンティクスが大きな役割を果たしている．ところが，従来のデータベースシステムでは，このような応用分野の性質を活用する事が困難である．特にバッファ管理手法に関して，従来のデータベースシステムで多く用いられてきた LRU[1] などの手法はこのような特殊な問い合わせに対して効果が高くない事が知られている．

このような背景に基づき，本研究では地理情報の特徴を活用したバッファ管理手法 KNC-D を提案する．KNC-D 法は，GIS が多くの場合空間オブジェクトを地理的近接関係に従って構造化し，管理している事を利用する．直観的には，KNC-D は，地理情報が地図上の領域 (Cells) 毎にクラスタ化されている事を仮定し，“アクセスを行う領域に距離的に近い領域を含むページは，なるべくバッファ中に保存する (**Keep Nearer**)” 戦略をとる．

2　GIS の問い合わせ例

KNC-D の直観的な理解のため，航行を伴う GIS 問い合わせの例を用いる．ここでは，地図 1上の問い合わせ「ある幹線道路 R に対して，最も近い不動産物件を検索せよ」を考える．簡単のため，空間索引構造等は利用できないものとし，空間オブジェクトとしての道路の情報は領域に応じた複数の断片に分割されて管理されているものとする．このとき，検索の実行の手順の一例として，

> R の断片の存在する領域を追跡しながら，その領域の周辺の領域に不動産物件が存在するかを検査し，存在した場合，その物件から R へ至る道程を計算する

といった方法があるが，この場合のアクセスパターンとして abdbacdfggbcfh$\cdots$ が考えられる．この例に示されるように航行を伴う問い合わせでは，ある領域に対するアクセスの後，それに隣接する領域に対するアクセスが起こりやすい．従って，領域間の位置関係をバッファ管理に利用する事は有効である．

KNC-D : A Buffer Management Policy for GIS
Junichi Notoya, Hanxiong Chen, Nobuo Ohbo
University of Tsukuba
1-1-1 Tennodai, Tsukuba, Ibaraki 305, Japan

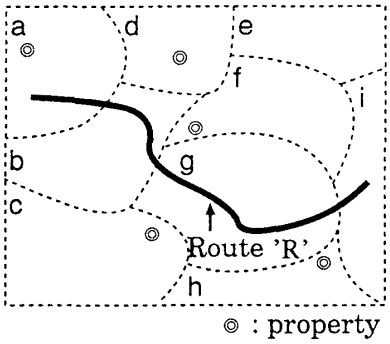

図 1: GIS 問い合わせ

KNC-D ではアクセスを行う領域とバッファ内の領域との間の距離を求め，その値が D より小さな領域を含むページについては，バッファ内での優先度を引き上げる．ページアウトが必要な場合は，優先度の小さなものを選択する．領域間の距離は，様々な定義が考えられるが，本研究では互いに隣接する領域の間の距離を 1 とする方法を取る．

先の例では，バッファサイズ 4 の場合，LRU が b の複数回の出現を利する事が出来ないのに対し，KNC-D では b が g に隣接するため，より高い効果を期待できる．

3　KNC-D アルゴリズム

GIS では地理情報の二次記憶への格納方法として，メッシュ構造 [2] を用いる方法，Quadtree[4] などの空間索引構造を用いる方法などが用いられる．これらはいずれも，地図を領域に分割し，各領域に含まれる空間オブジェクトを同一ページにクラスタ化する．バッファ管理に関する考察では，領域によってクラスタ化されている事と，各領域の近接関係が得られる事が重要であり，これらの格納方法のいずれを利用するかには依存しない．そこで，本稿では GIS における抽象化された地図を以下のように定義する．

定義 1 (地図) GIS における地図とは，グラフ $G = (N, E)$ である．N は領域の集合であり，$E \subseteq N \times N$ は領域間の隣接関係である．$(c_i, c_j) \in E$ を $c_i \sim c_j$ とも書く．

定義 2 (距離) 領域間の距離 $\mathrm{dist}(c_i, c_j)$ とは，

$$\mathrm{dist}(c_i, c_j) = \begin{cases} 0, & c_i = c_j, \\ \min\{\mathrm{dist}(c_i, c_k) | c_k \sim c_j\} + 1, & \exists c_k . c_k \sim c_j, \\ \infty, & otherwise. \end{cases}$$

また，アルゴリズムの呈示のために，以下のような表現を用いる．

定義 3 (バッファ状態) バッファ状態とは，定義域を N とする列 B である．$B|n (n \in N)$ は，バッファ状態 B において n へのアクセスが発生した場合の新しいバッファ状態を表す．$\pi_k(B)$ は B の先頭 k 個の要素からなる部分列を，$B\downarrow_p$ は述語 p を満たす要素を全て含む B の部分列を表す．

これらの定義を用い, 従来の LRU, LRU-K[3] と我々の提案する KNC-D の各々のアルゴリズムをバッファサイズ m の場合について示す.

LRU

$$B|n = n\pi_{m-1}(B\downarrow_{\lambda x.x\in B-\{n\}})$$

LRU-K

$$B|n = n\pi_{m-1}(\prod_{i=0}^{\infty} B\downarrow_{\lambda x.x\in B-\{n\}\wedge b_t(x,K)=i})$$

KNC-D

$$B|n = n\pi_{m-1}(B\downarrow_{\lambda x.x\in B-\{n\}\wedge \mathrm{dist}(x,n)\le D} B\downarrow_{\lambda x.x\in B-\{n\}\wedge \mathrm{dist}(x,n)>D})$$

ここで, LRU-K における $b_t(x,K)$ は, x の 'Backward K-distance[3]' を表す. また, LRU は LRU-K の $K=1$ の場合及び KNC-D の $D=0$ の場合に相当する.

さらに, アクセス列を以下のように定義する.

定義 4 (アクセス列) アクセス列とは, 定義域を N とする列 A であり,

1. $A=\varepsilon$(空列) は, アクセスを行わない事を示す.
2. A' がアクセス列, $n\in N$ のとき, $A=A'n$ は A' で示される動作の直後に n に対するアクセスを行う事を示す.

このとき, アクセス列 $A=a_1a_2\ldots a_n$ を実行した直後のバッファ状態は, $\varepsilon|a_1|a_2|\ldots|a_n$ である.

4 実験と評価

KNC-D アルゴリズムの評価のために, 以下のような仮定のもと, シミュレーションによる実験を行った. 実験に用いるアクセスパターンは, 2節の例で行った考察に従い, 以下のように定める.

定義 5 (Sequential Access on MAP) 地図上の Sequential access とは, アクセス列 $A^s=\prod_{i=0}^{I} a_i^0a_i^1\ldots a_i^{t_i}$ であり, 任意の i に対して $a_i^0\sim a_{i+1}^0$, また, 任意の $k_i(k_i=1,\ldots,t_i)$ に対して $a_i^0\sim a_i^{k_i}$ が成立する.

地図を 1000 × 1000 メッシュ構造, アクセス列の大きさを $|A^s|=10000$ とした実験の結果を図 2に示す.

図 2より, KNC-D は今回比較した他のアルゴリズムよりも常に高いヒット率を示す事がわかる. 一方, LRU-K はバッファが大きくなるとかえって比較的低いヒット率を示す. KNC-D は特にバッファの個数が 5 から 15 である場合に, 他のアルゴリズムとの差が顕著である。GIS では個々のバッファの容量, 即ちページサイズは一度に画面に呈示すべき範囲内の情報量に依存するが, 一般には関係データベースシステムなどにおけるバッファの容量よりも大きくならざるを得ないと考えられる. 従って, 同程度の一次記憶装置を持つシステムであれば, GIS のバッファの個数は関係データベースシステムと比較して少数となる. これより, GIS においては KNC-D は有効なアルゴリズムであるといえる.

また, KNC-D(D=1) と KNC-D(D=2) との差が KNC-D(D=1,D=2) と LRU との差と比較して微小である事から, KNC-D は D=1 で既に十分有効であると云える.

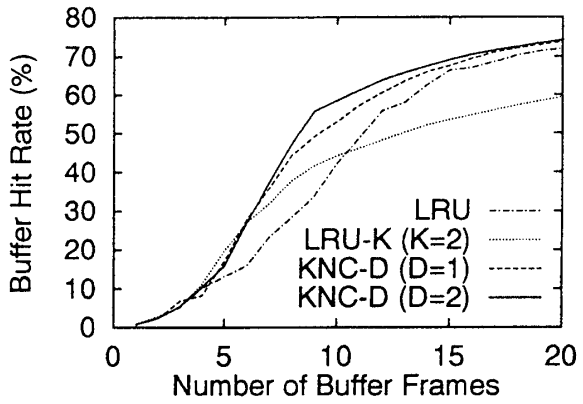

図 2: 1000 × 1000 メッシュ構造

5 まとめ

実験から, 今回我々が提案した KNC-D アルゴリズムが GIS におけるバッファ管理アルゴリズムとして有効であるという事が明らかになった.

また, LRU-K における K と同様に, 問い合わせの種類にしたがって D の値を動的に変える事により, 従来型の問い合わせにも対応できる事から, KNC-D は十分実用に耐えうる手法であると云える.

本手法の問題点は, ヒット率の上昇に伴う性能の向上に対して, 他の手法より余分にかかる計算コストを十分小さくする必要がある事である. 本手法において主な問題点となる dist() の計算量は, 一般には Dijkstra の方法などにより, $O(|N|^2)$ であるが,

$$\sum_{n\in N}|\{m|m\in N\wedge \mathrm{dist}(m,n)\le D\}| \ll |N|^2$$

となるように D を選べば, $O(|N|)$ とすることが可能であり, 特に, メッシュ, Quadtree などでは $O(1)$ にできる場合がある.

今後は解析的なコスト式の確立を行い, 各種パラメータと性能の関係を厳密に導く必要がある. また、関係データベースの分野で行われているようなアクセスパターンの標準化を GIS に対して行う事も重要な課題である。

参考文献

[1] Effelsberg, W. and Haerder, T.: Principles of Database Buffer Management, *ACM Trans. on Database Systems*, 9 No. 4, pp. 560–595 (1984).

[2] Goodchild, M. F. and Kemp, K. K. eds.: *Technical Issues in GIS*, NCGIA Core Curriculum, Univ. of California (1990).

[3] O'Neil, E. J., O'Neil, P. E., and Weikum, G.: LRU-K Page Replacement Algorithm For Database Disk Buffering, *ACM SIGMOD RECORD*, pp. 297–306 (1993).

[4] Samet, H.: The Quadtree and Relate Hierarchical Data Structures, *ACM Computing Surv.*, **16** No. 2, pp. 187–260 (1984).